<u>In the claims:</u>

1.   (currently amended) A computer system comprising:
        at least two CPUs;
        shared memory<u>, which is</u> shared by said <u>at least two</u>
CPUs; and
        at least one shared system resource accessible to
said <u>at least two</u> CPUs,
        said shared memory having therein a resource locking
table, comprising memory elements, each of said memory
elements designated for being written to by only one of
said at least two CPUs, ~~and~~
        each CPU <u>of said at least two CPUs</u> having a
corresponding memory element for each <u>of said one</u> shared
system resource to which it has access,
        wherein each of said at least two CPUs is
communicatively interconnected with said shared memory
and said <u>at least one</u> shared system resource, ~~and~~
        <u>wherein</u> said resource locking table is operative by
each of said <u>at least two</u> CPUs, and
        wherein said communicative interconnection is across
a communications bus ~~wherein~~ <u>providing</u> a single read
operation ~~is~~ capable of atomically reading ~~at least~~ a
collection of said memory elements, said collection
comprising at least two memory elements.

2. (deleted)

3. (deleted)

4. (deleted)

5.   (currently amended) A computer system comprising:

2

at least two CPUs;

shared memory, which is shared by said at least two CPUs; and

at least one shared system resource accessible to said at least two CPUs+,

said shared memory having therein a resource locking table, comprising memory elements, each of said memory elements designated for being written to by only one of said at least two CPUs, ~~and~~

each CPU of said at least two CPUs having a corresponding memory element for each of said at least one shared system resource to which it has access,

wherein each of said at least two CPUs is communicatively interconnected with said shared memory and said at least one shared system resource, ~~and~~

wherein said resource locking table is operative by each of said at least two CPUs, and ·

wherein any of said at least two CPUs can read a first collection of memory elements in a single transaction, said first collection of memory elements corresponding to requests ~~of~~ received from said at least two CPUs for one of said at least one shared system ~~resources~~ resource.

6. (currently amended) The computer system according to claim 5, wherein a CPU of said at least two CPUs locks ~~a~~ said at least one shared system resource by executing at least one control ~~commands~~ command to accomplish the steps of:

a. checking a content of said first collection of memory elements of said shared memory and if each memory element of said first collection of memory elements is not clear, waiting for each of said first collection of memory elements to clear;

3

b. setting ~~the~~ said corresponding memory element ~~corresponding to~~ for each CPU of said ~~CPU~~ at least two CPUs;

c. checking if more than one of said memory elements of said first collection of memory elements is set, and if true then resetting said corresponding memory element ~~corresponding to~~ for each CPU of said ~~CPU~~ at least two CPUs and waiting for each of said memory elements of said first collection of memory elements to reset and repeating the sequence from step a.;

d. accessing said at least one shared system resource corresponding to said first collection of memory elements; and

e. resetting said corresponding memory element, corresponding to said ~~CPU~~ at least two CPUs, of said first collection of memory elements corresponding to said at least one shared system resource.

7. (currently amended) The computer system according to claim 5, wherein a second collection of memory elements corresponds to a group of CPUs.

8. (currently amended) The computer system according to claim 7, wherein ~~a~~ said CPU group locks ~~e~~ said at least one shared system resource by executing at least one control ~~commands~~ command to accomplish the steps of:

a. checking a first content of said second collection of memory elements and, if not clear, waiting for each memory element of said second collection of memory elements to clear;

b. setting ~~the~~ said memory element of said second collection of memory elements corresponding to ~~a desired~~ said CPU group;

4

c. checking if more than one of said memory elements of said second collection of memory elements is set, and if true then resetting said second collection of memory element elements corresponding to said CPU group and waiting for each of said memory elements of said second collection of memory elements to reset and repeating the sequence from step a.;

d. checking a second content of said first collection of memory elements corresponding to said at least one shared system resource and said CPU group and if not clear, waiting for each memory element of said first collection of memory elements to clear;

e. setting the said corresponding memory element of said first collection of memory elements corresponding to a CPU within said CPU group;

f. checking if more than one, of said memory elements of said first collection of memory elements is set, and if true then resetting said second collection of memory element elements corresponding to said CPU group and waiting. for each memory element of said memory elements of said first collection of memory elements to reset and repeating the sequence from step a.;

g. accessing said at least one shared system resource corresponding to said first collection of memory elements;

h. resetting said corresponding memory element, corresponding to said CPU group, of said first collection of memory elements corresponding to said at least one shared system resource; and

i. resetting said corresponding memory element, corresponding to said CPU group, of said second collection of memory elements corresponding to said at least one shared system resource.

9. (deleted)

5

10. (currently amended) A method for locking a shared system
    resource for use by a single CPU, the method comprising
    the steps of:

    a. checking a content of a collection of memory
    elements of a resource locking table implemented in
    shared memory and if each memory element of said
    collection of memory elements is not clear, waiting for
    each of said collection of memory elements to clear;

    b. setting ~~the~~ said corresponding memory element of
    said collection of memory elements corresponding to the
    single CPU;

    c. checking if more than one of said memory elements
    of said collection of memory elements is set, and if true
    then resetting said corresponding memory element
    corresponding to the single CPU and waiting for each of
    said memory elements of said collection of memory ~~units~~
    elements to reset and repeating the sequence said from
    step a;

    d. accessing ~~the~~ said shared system resource
    corresponding to said collection of memory elements; and

    e. resetting ~~the~~ said corresponding memory element,
    corresponding to the single CPU, of said collection of
    memory elements corresponding to said shared system
    resource.

11. (previously amended) The method according to claim 10,
    wherein said shared memory is connected to a PCI bus.

12. (currently amended) A method for locking a shared system
    resource for use by a single CPU having access to a first
    and a second collection of memory elements of shared
    memory, the method comprising the steps of:

6

a. checking ˍa <u>first</u> content of said second collection of memory elements and if not clear, waiting for each memory element of said second collection of memory elements to clear;

b. setting ~~the~~ <u>a corresponding</u> memory element of said second collection of memory elements corresponding to a ~~desired~~ CPU group;

c. checking if more than one of said memory elements of said second collection of memory elements is set, and if true, resetting said <u>corresponding</u> memory element corresponding to said CPU group and waiting for each of said memory elements of said second collection of memory elements to reset and repeating ~~the sequence~~ <u>said</u> <u>checking step</u> ~~from~~ <u>of</u> step a.;

d. checking a <u>second</u> content of said first collection of memory elements corresponding to said <u>shared system</u> resource and said CPU group and, if not clear, waiting for each memory element of said first collection of memory elements to clear;

e. setting said <u>corresponding</u> memory element of said first collection of memory elements corresponding to ~~a CPU within~~ said CPU group;

f. checking if more than one of said memory elements of said first collection of memory elements is set, and if true then resetting said <u>corresponding</u> memory element corresponding to said CPU <u>group</u> and waiting for each of said memory elements of said first collection of memory elements to reset and repeating ~~the sequence~~ <u>said</u> <u>checking step</u> ~~from~~ <u>of</u> step a.;

g. accessing ~~the~~ <u>said</u> shared system resource corresponding to said first collection of memory elements;

h. resetting said <u>corresponding</u> memory element, corresponding to said CPU <u>group,</u> of said first collection

7

of memory elements corresponding to said shared system resource; and

    i. resetting said <u>corresponding</u> memory element, corresponding to said CPU group, of said second collection of memory elements corresponding to said shared system resource.

13. (previously amended) The method according to claim 12, wherein said shared memory is connected to a PCI bus.

14. (currently amended) A computer program product for locking a shared system resource for use by a single CPU, the computer program product comprising:

    computer readable medium having thereon software instructions for enabling a system, containing at least two CPUs and at least one shared system resource, to perform predetermined operations comprising <u>the steps of</u>:

    a. checking a content of a collection of memory elements of a resource locking table implemented in shared memory and if each memory element of said collection of memory elements is not clear, waiting for each of said <u>collection of</u> memory elements to clear;

    b. setting ~~the~~ <u>each</u> memory element <u>of said</u> <u>collection of memory elements</u> corresponding to ~~said~~ <u>the</u> <u>single</u> CPU;

    c. checking if more than one of said memory elements of said collection of memory elements is set, and if true then resetting ~~said~~ <u>each</u> memory element <u>of said</u> <u>collection of memory elements</u> corresponding to ~~said~~ <u>the</u> <u>single</u> CPU and waiting for each of said <u>collection of</u> memory elements ~~of said collection of memory units~~ to reset and repeating the sequence ~~said~~ from step a.;

    d. accessing said shared system resource corresponding to said collection of memory elements; and

8

e. resetting ~~said~~ each memory element of said corresponding memory elements, corresponding to ~~said~~ the single CPU, of said collection of memory elements corresponding to said shared system resource.

15. (currently amended) The computer ~~software~~ program product according to claim 14, wherein said ~~CPUs~~ at least two CPUs and said shared memory are connected by a PCI bus.

16. (currently amended) A computer program product for locking a shared system resource for use by a single CPU having access to a first and a second collection of memory elements of at least one shared memory, the computer program product comprising:

computer readable medium having thereon software instructions for enabling a system, containing at least two CPUs and at least one shared system resource, to perform predetermined operations, comprising the steps of:

a. checking a first content of said second collection of memory elements and if not clear, waiting for each memory element of said second collection of memory elements to clear;

b. setting ~~the~~ each memory element of said second collection of memory elements corresponding to a ~~desired~~ CPU group;

c. checking if more than one of said memory elements of said second collection of memory elements is set, and if true then resetting ~~said~~ each memory element corresponding to said CPU group and waiting for each memory element of said ~~memory elements of said~~ second collection of memory elements to reset and repeating the sequence from step a.;

9

d. checking a <u>second</u> content of said first collection of memory elements residing in a resource locking table implemented in said <u>at least one</u> shared memory and further corresponding to said shared system resource and <u>said</u> CPU group and, if not clear, waiting for each <u>memory element</u> of said ~~memory elements of said~~ first collection of memory elements to clear;

e. setting ~~said~~ <u>each</u> memory element of said first collection of memory elements corresponding to ~~a CPU within~~ said CPU group;

f. checking if more than one <u>memory element</u> of said ~~memory elements of said~~ first collection of memory elements is set, and if true then resetting ~~said~~ <u>each</u> memory element corresponding to said CPU <u>group</u> and waiting for each <u>memory element</u> of said ~~memory elements of said~~ first collection of memory elements to reset and repeating the sequence from step a.;

g. accessing ~~the~~ <u>said</u> shared system resource corresponding to said first collection of memory elements;

h. resetting ~~the~~ <u>each</u> memory element, corresponding to said CPU <u>group</u>, of said first collection of memory elements corresponding to said shared system resource; and

i. resetting ~~said~~ <u>each</u> memory element, corresponding to said CPU group, of said second collection of memory elements corresponding to said shared system resource.

17. (currently amended) The computer ~~software~~ program product according to claim 16, wherein said <u>at least two</u> CPUs and said <u>at least one</u> shared memory are connected by a PCI bus.

10